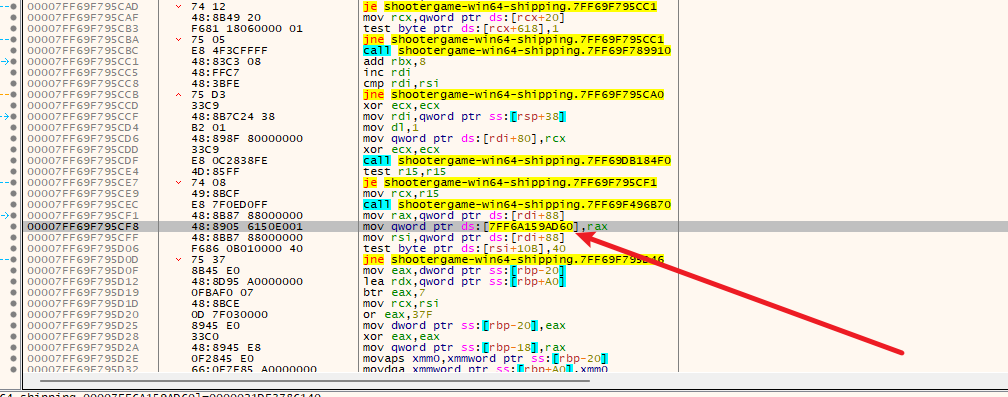
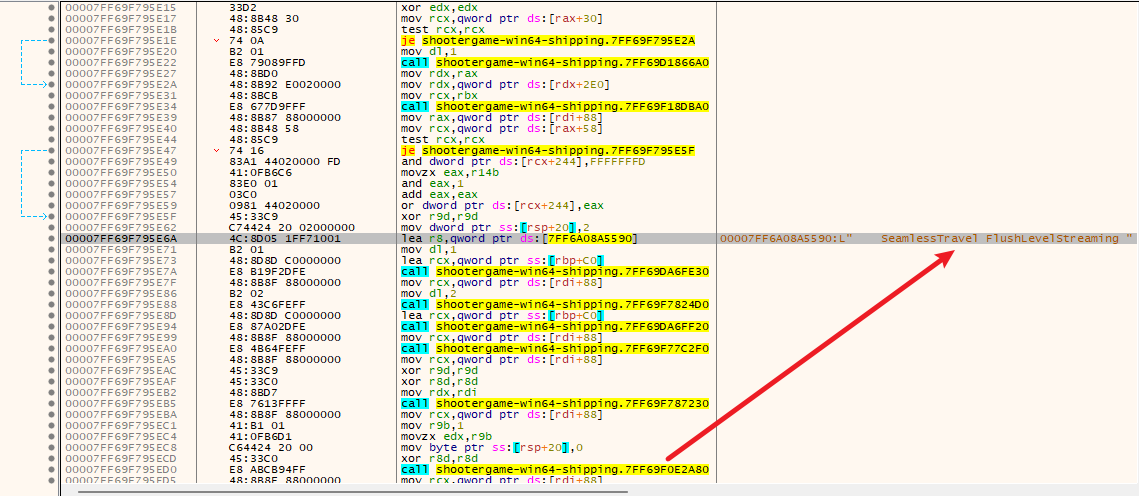
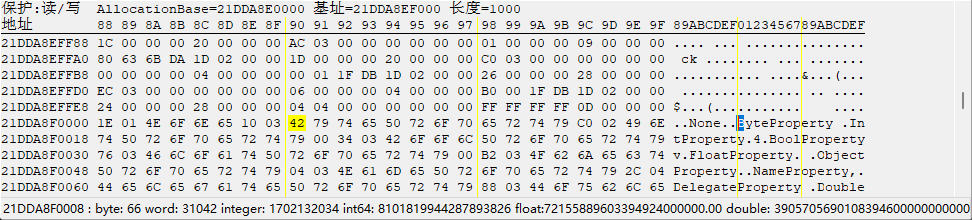
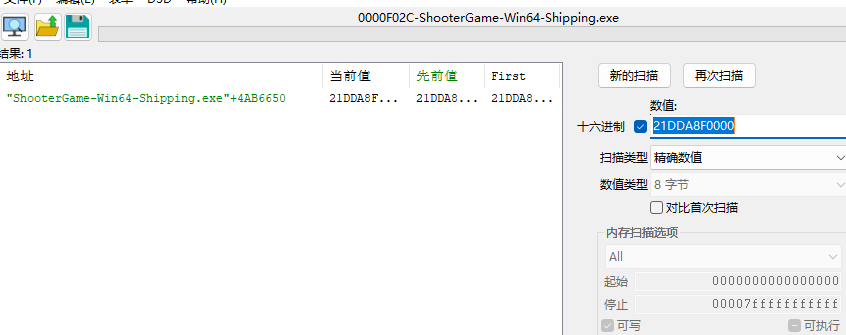
1. 寻找游戏实例中UWorld、GName、GObject的动态地址，方便后续计算偏移
2. 利用UWorld的特征码”seamlessTravel FlushLevelStreaming”，用xdbg附加进程搜索主模块字符串，然后定位到UWorld的动态地址所在处，记录下UWorld的动态地址0x7FF6A159AD60

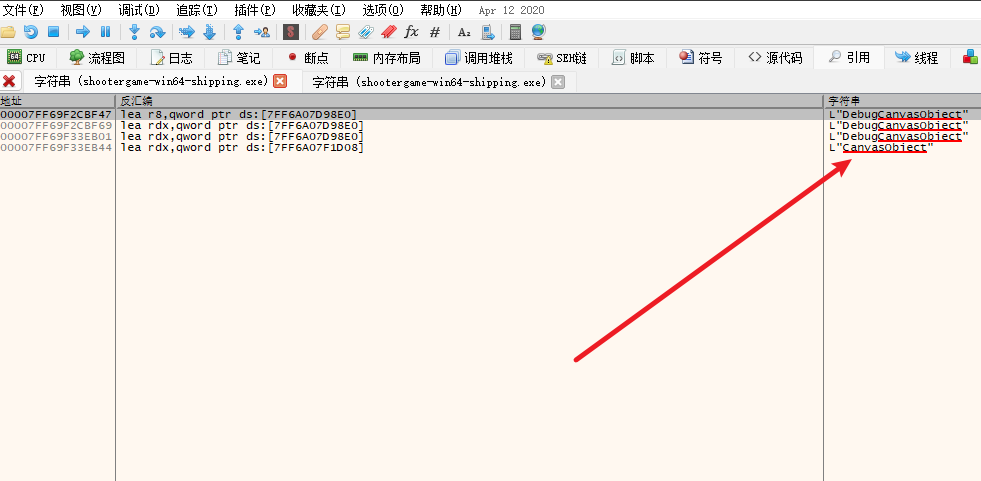


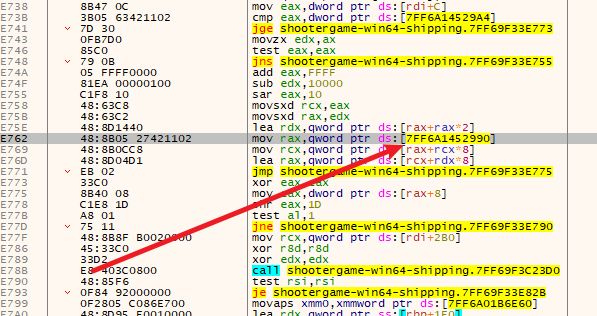
1. 用CE搜索字符串ByteProperty去定位GName，ByteProperty的地址减0x8就是None的地址，去搜索十六进制，得到None的静态地址，用None的地址减0x10即GName的地址，即"ShooterGame-Win64-Shipping.exe"+4AB6650-0x10





1. 用xdbg搜索字符串canvasobject，定位GObject，找到GObject的动态地址0x00007FF6A1452990





1. 计算UWorld、GName、GObject的静态地址

由CE易得主模块基地址：0x7FF69C960000

UWorld

偏移：0x7FF6A159AD60 - 0x7FF69C960000 = 0x4C3AD60

静态： "ShooterGame-Win64-Shipping.exe" + 0x4C3AD60

GName

静态："ShooterGame-Win64-Shipping.exe" + 4AB6650 - 0x10

GObject

偏移：0x00007FF6A1452990 - 0x7FF69C960000 = 4AF 2990

静态："ShooterGame-Win64-Shipping.exe" + 4AF 2990

1. 利用得到的GName偏移与GObject偏移去dump SDK



1. 关键数据及结构体
2. 骨骼坐标

UWorld结构的0x30处有关卡结构指针：ULevel\* PersistentLevel

ULevel结构的0x98处有世界中所有对象的数组：TArray<AActor\*> Actors

AActor→APawn→ACharacter（箭头代表派生关系）

ACharacter结构中有骨骼网格组成： USkeletalMeshComponent\* Mesh

USkeletalMeshComponent结构的0x4B0处有所有骨骼转换信息的数组：TArray<FTransform> BoneTransform，骨骼坐标在内存中为相对坐标，需要利用矩阵转换才能转换为表示世界信息的矩阵，包含世界坐标。

1. 人物ID

UWorld结构的0x120处有游戏状态指针：AGameStateBase\* GameState

AGameStateBase结构的0x238处有玩家状态的指针数组：TArray<APlayerState\*> PlayerArray;

APlayerState结构的0x300处有角色名称的FString结构：FString PlayerNamePrivate

FString结构的第一成员就是TCHAR类型的字符数组：TArray<TCHAR> DataType

1. 利用虚表Hook去拦截ue引擎绘制函数Present的执行，并利用ImGui在MyPresent中添加我们自己的绘制逻辑，最后将我们生成的dll注入到游戏进程，就可完成cheat。

